



Ciências
ULisboa

Modelação Numérica 2022

Aula 7

Discretização. Equações diferenciais às derivadas parciais. Diferenças finitas. Equação de advecção linear.

Discretização de uma função contínua 4D

$$V = V(x, y, z, t)$$

$$V_{i,j,k}^n = V(x_0 + i\Delta x, y_0 + j\Delta y, z_0 + k\Delta z, t_0 + n\Delta t)$$

$$i = 0, \dots, N_x - 1; j = 0, \dots, N_y - 1; k = 0, \dots, N_z - 1; n = 0, \dots, N_t - 1$$

Resolução espaço-temporal

$$\Delta x, \Delta y, \Delta z, \Delta t$$

Domínio

$$x \in [x_0, x_0 + (N_x - 1)\Delta x], y \in [y_0, y_0 + (N_y - 1)\Delta x], z \in [z_0, z_0 + (N_z - 1)\Delta x]$$
$$t \in [t_0, t_0 + (N_t - 1)\Delta t]$$

Características da DISCRETIZAÇÃO

(Resolução) **Intervalo de amostragem** (constante) < metade do período (tempo), ou metade comprimento de onda (série no espaço).

Fase de amostragem: localização da primeira amostra

(Domínio) **Truncatura**: dimensão da amostra (Número de pontos)

Erro de arredondamento: precisão da representação de números “reais” (DOUBLE)

Equações diferenciais às derivadas parciais

Muitos problemas envolvem a solução de equações diferenciais. A sua solução numérica requer **discretização** e sua transformação em equações algébricas: o **teorema da amostragem** deve ser satisfeito.

No caso das equações diferenciais às derivadas parciais existem 2 ou mais **variáveis independentes**, podendo uma dessas variáveis ser o tempo.

É conveniente classificar os problemas representados por estas equações em três classes: **problemas de condições iniciais**, **problemas de condições fronteira** (espaciais), **problemas mistos**.

Equações (exemplos com primeiras e segundas derivadas)

Equação de **advecção** (linear, 1D) $\frac{\partial T}{\partial t} = -u \frac{\partial T}{\partial x}, u = \text{const}$

Equação da **difusão** (linear, 1D) $\frac{\partial T}{\partial t} = K \frac{\partial^2 T}{\partial x^2}, K = \text{const}$

Equação de **Poisson** (2D) $\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} = f(x, y)$

Equação de **Laplace** (2D) $\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$

Equação de **ondas** (2D) $\frac{\partial^2 u}{\partial t^2} = c^2 \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$

Equação de Navier-Stokes

$$\frac{\partial u}{\partial t} = \underbrace{-u \frac{\partial u}{\partial x} - v \frac{\partial u}{\partial y} - w \frac{\partial u}{\partial z}}_{\text{Advecção não linear}} - \underbrace{\frac{1}{\rho} \frac{\partial p}{\partial x}}_{\text{Força do gradiente de pressão}} + \underbrace{fv}_{\text{Coriolis}} + \underbrace{\nu \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)}_{\text{Difusão}}$$

Equação fundamental da mecânica de fluidos.

É uma equação diferencial **não linear** de segunda ordem.

A **solução numérica** destas equações requer a sua transformação em **equações algébricas discretas**. Existem vários métodos: **diferenças finitas**, elementos finitos, método espectral.

As equações da Meteorologia:

$$\frac{\partial \vec{v}}{\partial t} = -(\vec{v} \cdot \nabla) \vec{v} + \vec{g} - \frac{1}{\rho} \nabla p + \eta \nabla^2 \vec{v} - 2\vec{\Omega} \times \vec{v} + \nu \nabla^2 (\vec{v}) \quad \text{Navier-Stokes (3)}$$

$$\frac{\partial \theta}{\partial t} = -(\vec{v} \cdot \nabla) \theta + Q_{\text{Radiativo}} + Q_{\text{Latente}} + \kappa \nabla^2 \theta \quad \text{Termodinâmica}$$

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \vec{v}) \quad \text{Continuidade}$$

$$\frac{\partial q_{v,l,g}}{\partial t} = -(\vec{v} \cdot \nabla) q_{v,l,g} + \text{Transição de fase} + \kappa_D \nabla^2 q_{v,l,g} \quad \text{Fases da água (3)}$$

$$p = R\rho T(1 + 0.61q_v) \quad \text{Equação de estado}$$

$$\theta = T \left(\frac{p}{p_{00}} \right)^{-k} \quad \text{Temperatura potencial}$$

✓ 10 equações
✓ 10 incógnitas
Bem posto

As equações da Meteorologia (e muitas outras)

Não têm solução analítica...

Não existe nenhum modelo analógico que as represente na sua totalidade: mas podem existir modelos interessantes para certos problemas particulares.

Também **não é possível escrever um modelo numérico completamente geral.**

... é preciso assumir compromissos!

Diferenças finitas (recordando...)

$$f(x_0 + \Delta x) = f(x_0) + \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 + \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots$$

$$f(x_0 - \Delta x) = f(x_0) - \left(\frac{\partial f}{\partial x}\right)_{x=x_0} \Delta x + \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x^2 - \frac{1}{3!} \left(\frac{\partial^3 f}{\partial x^3}\right)_{x=x_0} \Delta x^3 + \dots$$

$$\left(\frac{\partial f}{\partial x}\right)_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} - \frac{1}{2} \left(\frac{\partial^2 f}{\partial x^2}\right)_{x=x_0} \Delta x + \dots = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x} + \mathbf{O}(\Delta x)$$

$$\mathbf{Diferença centrada} \quad \left(\frac{\partial f}{\partial x}\right)_{x=x_0} = \frac{f(x_0 + \Delta x) - f(x_0 - \Delta x)}{2\Delta x} + \mathbf{O}(\Delta x^2)$$

2ª ORDEM

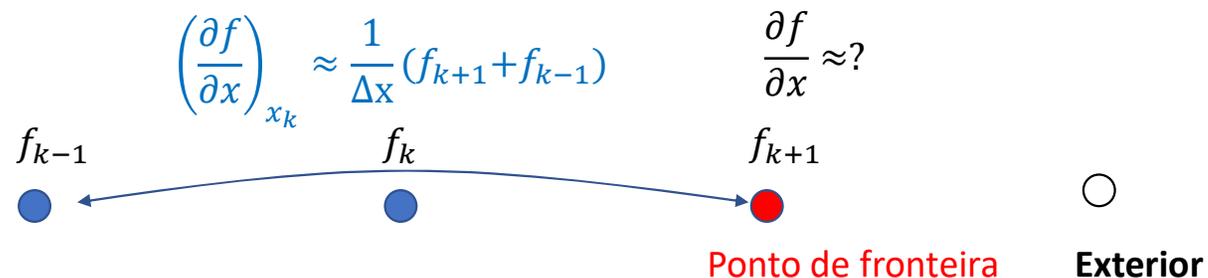
**Diferença
avançada
1ª ordem**

Resultado do cancelamento entre termos

Diferenças de ordem mais elevada

Combinando series de Taylor para $f(x_0 \pm n\Delta x)$, podem estabelecer-se aproximações às derivadas **até qualquer ordem**, implicando no entanto que o cálculo de uma derivada requer valores da função numa vizinhança cada vez mais extensa, o que não é prático e traz problemas quando nos aproximamos da **fronteira**.

Por exemplo, no caso das diferenças centradas, elas só podem ser calculadas em pontos interiores, devendo os pontos de fronteira ser calculados à parte (na condição fronteira)



Considerações iniciais

A **precisão** do método utilizado na discretização é só uma das propriedades relevantes a considerar.

Seja qual for a precisão, um método **consistente** deve convergir para a solução analítica no limite $\Delta x \rightarrow 0$.

Independentemente da precisão, um método só é útil se for numericamente **estável**, i.e. se o erro não crescer exponencialmente. Este critério é muito relevante para problemas que evoluem no tempo.

O erro de um método precisa de ser caracterizado em detalhe. Por exemplo: como se traduz na representação da propagação de ondas (**velocidade de fase** e de **grupo**), como discrimina os diferentes comprimentos de onda (**dispersão**), etc..

O **teorema da amostragem** é relevante! Estes tópicos serão importantes nos exemplos deste curso.

Equação de **advecção** (linear, 1D) $\frac{\partial T}{\partial t} = -u \frac{\partial T}{\partial x}$

A equação é **linear** se $u = \text{const}$ e tem, nesse caso, **solução analítica**. Em qualquer caso trata-se de um **problema de valores iniciais**. I.e., dada a distribuição inicial $T(x, t = 0)$ calcular $T(x, t, t > 0)$.

Vamos discretizar a função $T(x, t) \approx T_{k\Delta x}^{n\Delta t} \equiv T_k^n$ (O índice superior representa tempo, o inferior espaço). Vamos ensaiar uma solução por diferenças finitas usando o **método de Euler**, com diferenças avançadas no tempo e centradas no espaço:

$$\frac{T_k^{n+1} - T_k^n}{\Delta t} = -u \frac{T_{k+1}^n - T_{k-1}^n}{2\Delta x} \Rightarrow T_k^{n+1} = T_k^n - u\Delta t \frac{T_{k+1}^n - T_{k-1}^n}{2\Delta x}$$

Trata-se de um método com **1 nível** (o cálculo da solução no passo de tempo n só depende de 1 passo anterior $n - 1$). Trata-se de um **método explícito** T_k^{n+1} depende do campo no passo de tempo anterior (e não do seu valor noutros pontos em $t = n\Delta t$)

$$T_k^{n+1} = T_k^n - u\Delta t \frac{T_{k+1}^n - T_{k-1}^n}{2\Delta x}$$

Método Forward in Time Centred in Space (FTCS)

Trata-se de um método de 1ª ordem no tempo e de segunda ordem no espaço.

A solução depende do **estado inicial** e de **condições fronteira espaciais (problema misto)**.

Vamos definir a solução num domínio espacial finito

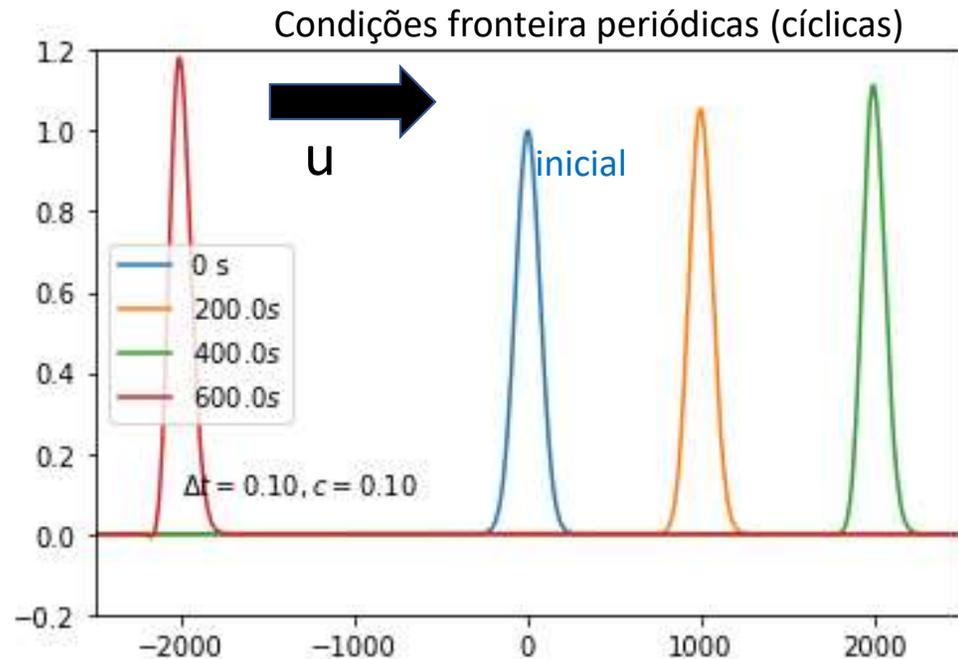
$$x \in [0, L_x] \Rightarrow x_k = k\Delta x, k = 0 \dots N - 1$$

Vamos considerar dois casos:

- Condições **cíclicas** (periódicas) $x_0 = x_N, x_{N+1} = x_1, etc.$
- Condições **“abertas”** $\frac{\partial T}{\partial x} = 0, em x = x_1, x = x_N$

Forward in Time Centred in Space FTCS (método instável)

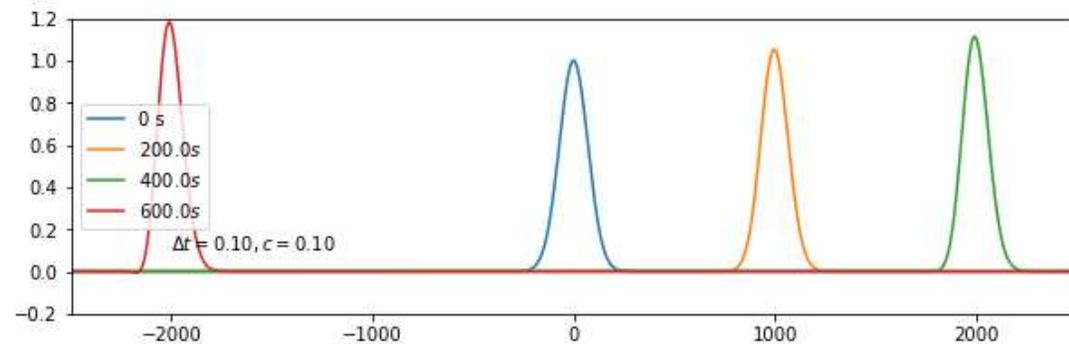
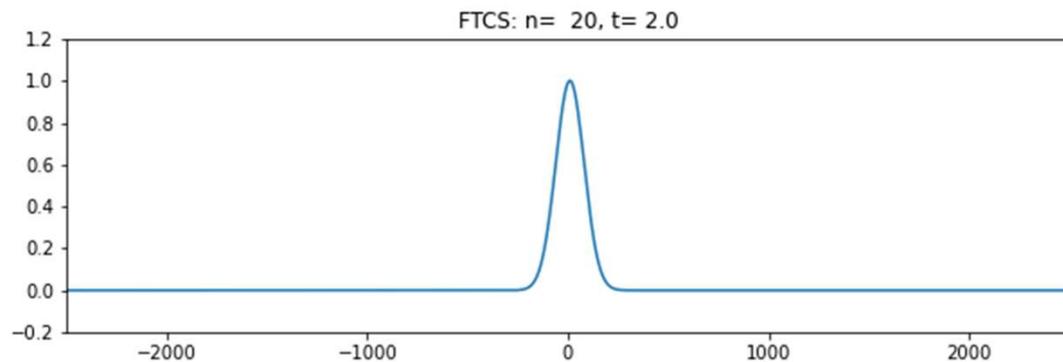
```
import numpy as np
import matplotlib.pyplot as plt
nx=1000;dx=5;Lx=nx/2*dx;u=5;dt=0.1
x=np.arange(-Lx,Lx,dx);nvolta=int(2*Lx/u/dt)
nt=nvolta//2+nvolta//4+1;passo=nvolta//5
courant=u*dt/dx
L=100;x0=0;TI=np.exp(-(x-x0)/L)**2)
T=np.copy(TI);TP=np.copy(T);
plt.plot(x,T,label='0 s')
plt.text(-2000,0.1,r'\Delta t=%3.2f, c=%3.2f $' % (dt,courant))
plt.axis([-Lx,Lx,-0.2,1.2])
for it in range(1,nt):
    for ix in range(1,nx-1):
        TP[ix]=T[ix]-u*dt/(dx*2)*(T[ix+1]-T[ix-1])
    TP[nx-1]=T[nx-1]-u*dt/(dx*2)*(T[0]-T[nx-2]) #fronteira cíclica
    TP[0]=T[0]-u*dt/(dx*2)*(T[1]-T[nx-1]) #fronteira cíclica
    T=np.copy(TP) #update temporal
    if it%passo==0:
        plt.plot(x,T,label=r'$%4.1f s$' % (dt*it))
        plt.axis([-Lx,Lx,-0.2,1.2])
plt.legend()
```



Se tentássemos estender a simulação mais $200 \Delta t$, o resultado “explodia” (erro cresce exponencialmente)

Agora com um filme...

```
...
import imageio
import os
...
movie=20
frames=[]
for it in range(1,nt):
    ...
    if it%movie==0:
        plt.figure(2,figsize=(10,3))
        plt.plot(x,T);
        plt.axis([-Lx,Lx,-0.2,1.2])
        plt.title('FTCS: n=%4i, t=%4.1f' %(it,dt*it))
        fn='mov'+str(it)+'.png'
        plt.savefig(fn); plt.clf()
        frames.append(fn)
...
images=[]
for frame in frames:
    images.append(imageio.imread(frame))
    os.remove(frame)
imageio.mimsave('FTCS'+'.gif',images,duration=0.1)
```



O método de Euler (FTCS) é **incondicionalmente** instável

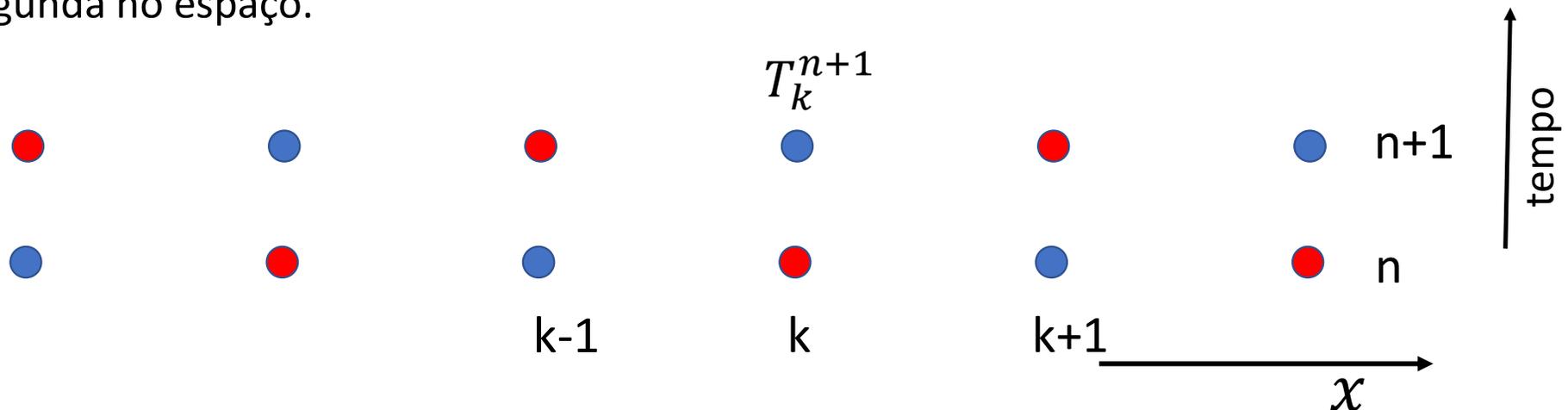
A instabilidade é neste caso **independente** da escolha dos parâmetros de discretização ($\Delta t, \Delta x$) e não é uma consequência da ordem da aproximação. É possível definir esquemas de 1ª ordem (ou ordem mais elevada) condicionalmente estáveis.

Aproximação de Lax

Em vez de $T_k^{n+1} = T_k^n - u\Delta t \frac{T_{k+1}^n - T_{k-1}^n}{2\Delta x}$

Fazemos: $T_k^{n+1} = \frac{1}{2}(T_{k-1}^n + T_{k+1}^n) - u\Delta t \frac{T_{k+1}^n - T_{k-1}^n}{2\Delta x}$

Continua a ser um método com 1 nível temporal e de primeira ordem no tempo e segunda no espaço.



LAX

```
...
for it in range(1,nt):
    for ix in range(1,nx-1):
        TP[ix]=0.5*(T[ix-1]+T[ix+1])-u*dt/(dx*2)*(T[ix+1]-T[ix-1])
    TP[nx-1]=0.5*(T[nx-2]+T[0])-u*dt/(dx*2)*(T[0]-T[nx-2]) #fronteira cíclica
    TP[0]=0.5*(T[1]+T[nx-1])-u*dt/(dx*2)*(T[1]-T[nx-1]) #fronteira cíclica
    T=np.copy(TP) #update temporal
```

...

FTCS

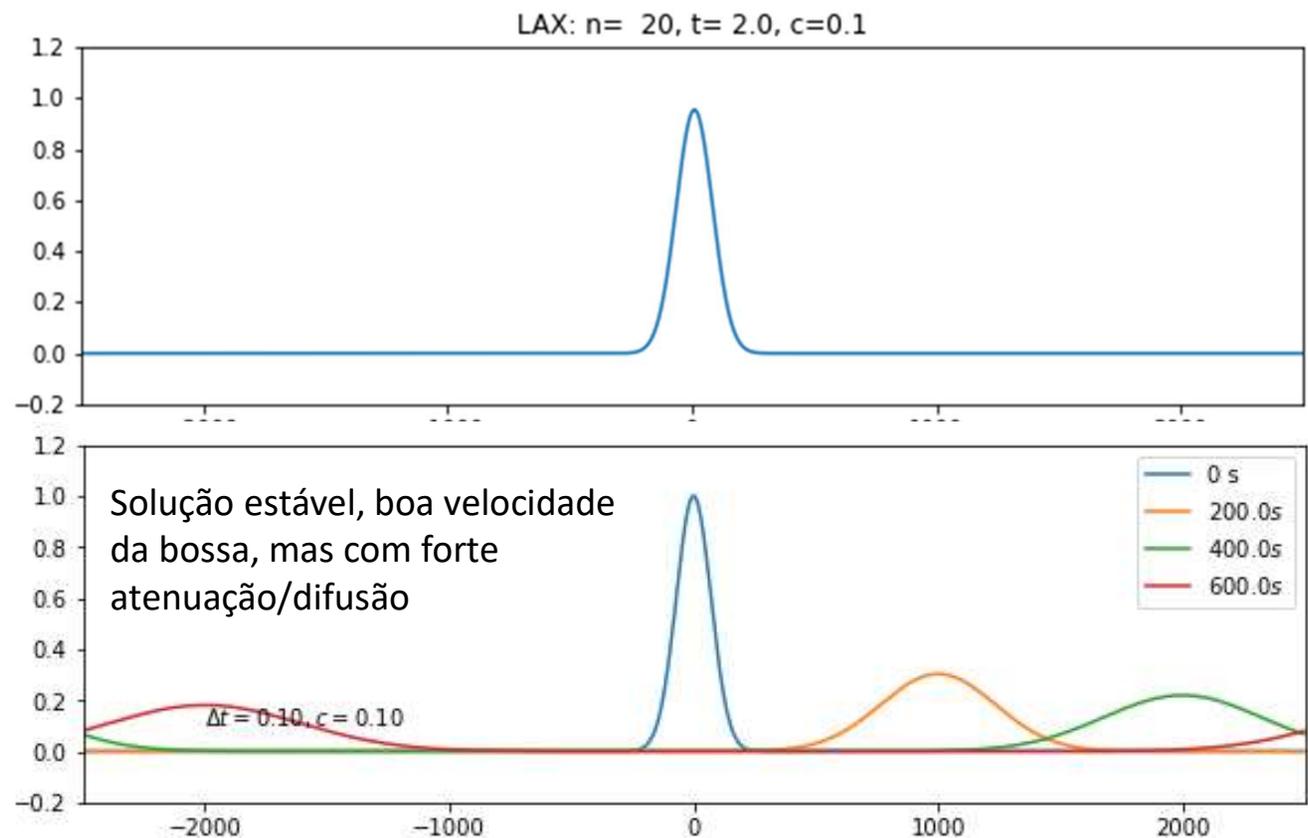
```
for it in range(1,nt):
    for ix in range(1,nx-1):
        TP[ix]=T[ix]-u*dt/(dx*2)*(T[ix+1]-T[ix-1])
    TP[nx-1]=T[nx-1]-u*dt/(dx*2)*(T[0]-T[nx-2]) #fronteira cíclica
    TP[0]=T[0]-u*dt/(dx*2)*(T[1]-T[nx-1]) #fronteira cíclica
    T=np.copy(TP) #update temporal
```

LAX Courant=0.1: difusão numérica

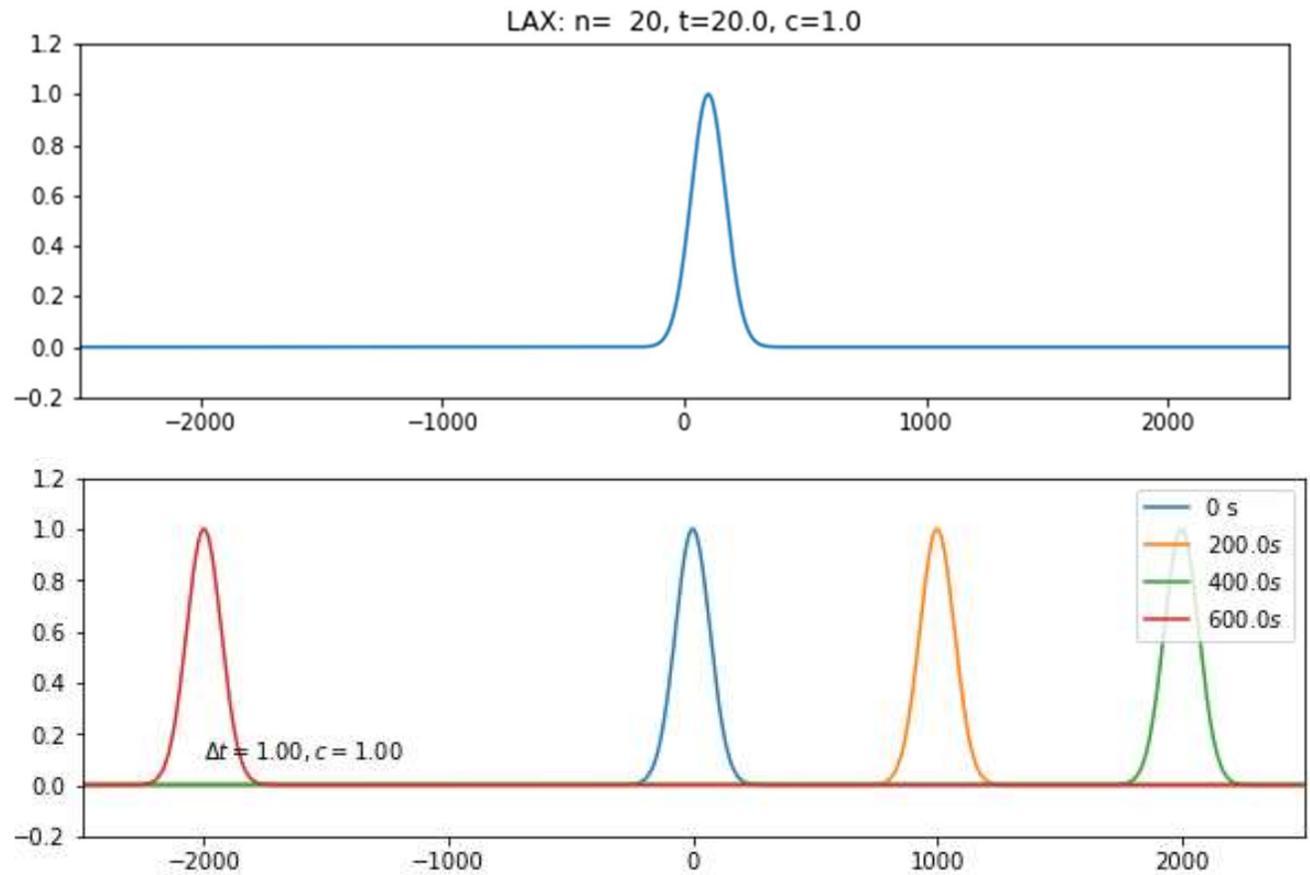
O comportamento do método depende do Número de Courant (of de Courant–Friedrichs–Lewy, CFL)

$$c = \frac{u\Delta t}{\Delta x}$$

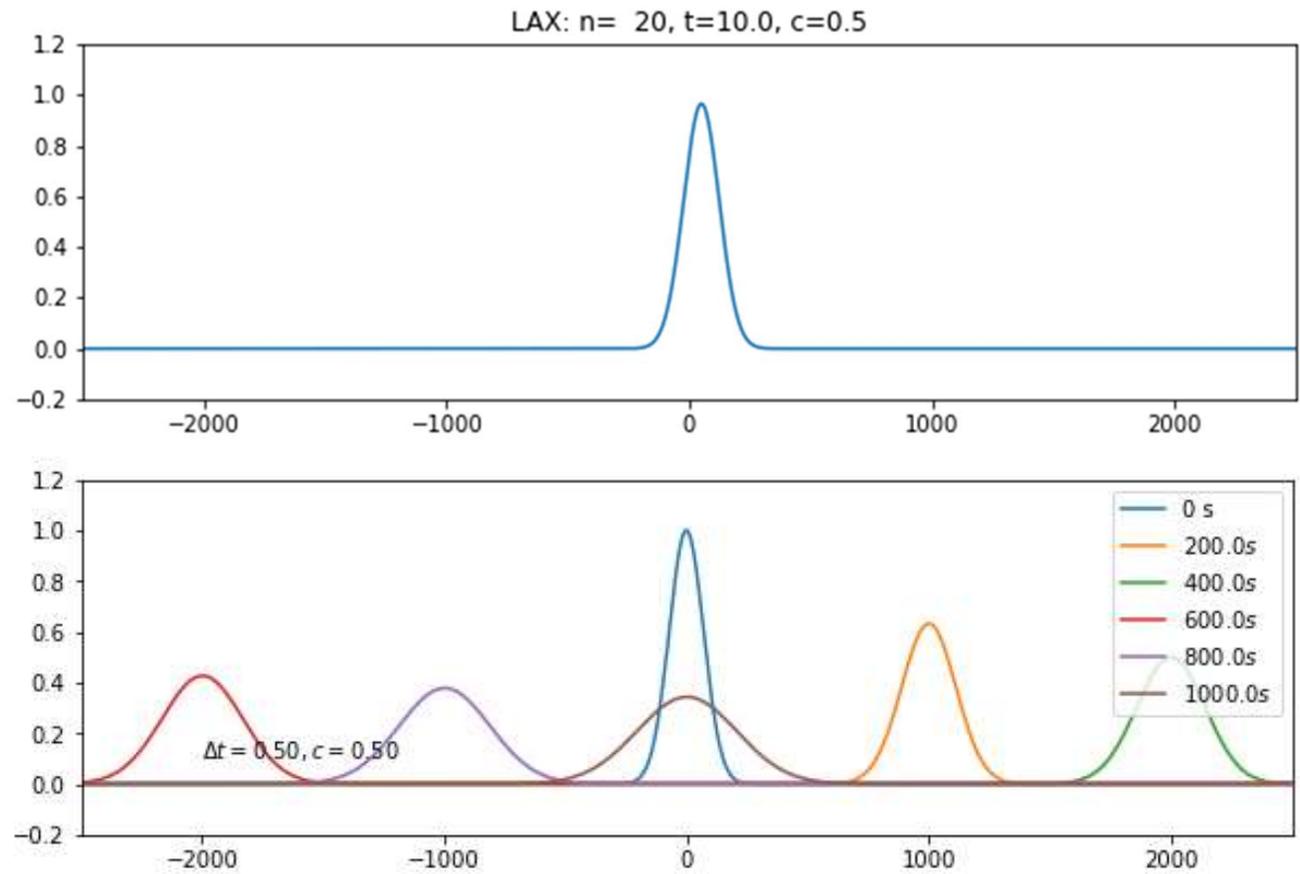
Notar que c é adimensional.



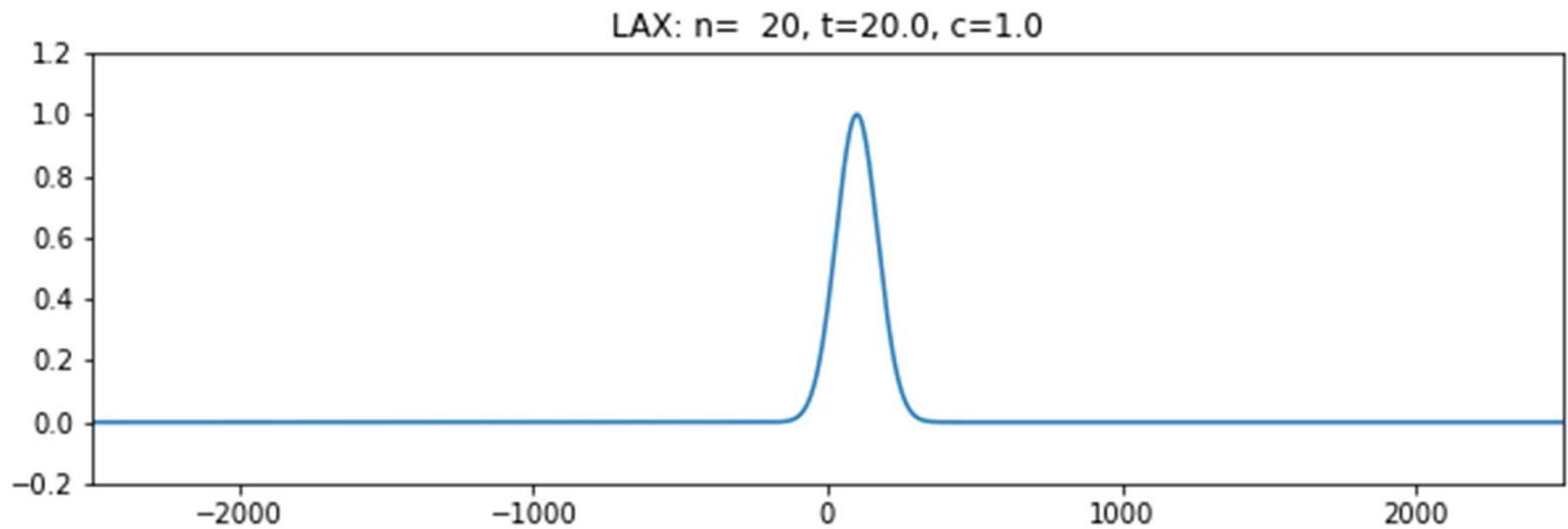
LAX Courant=1.0: sem atenuação



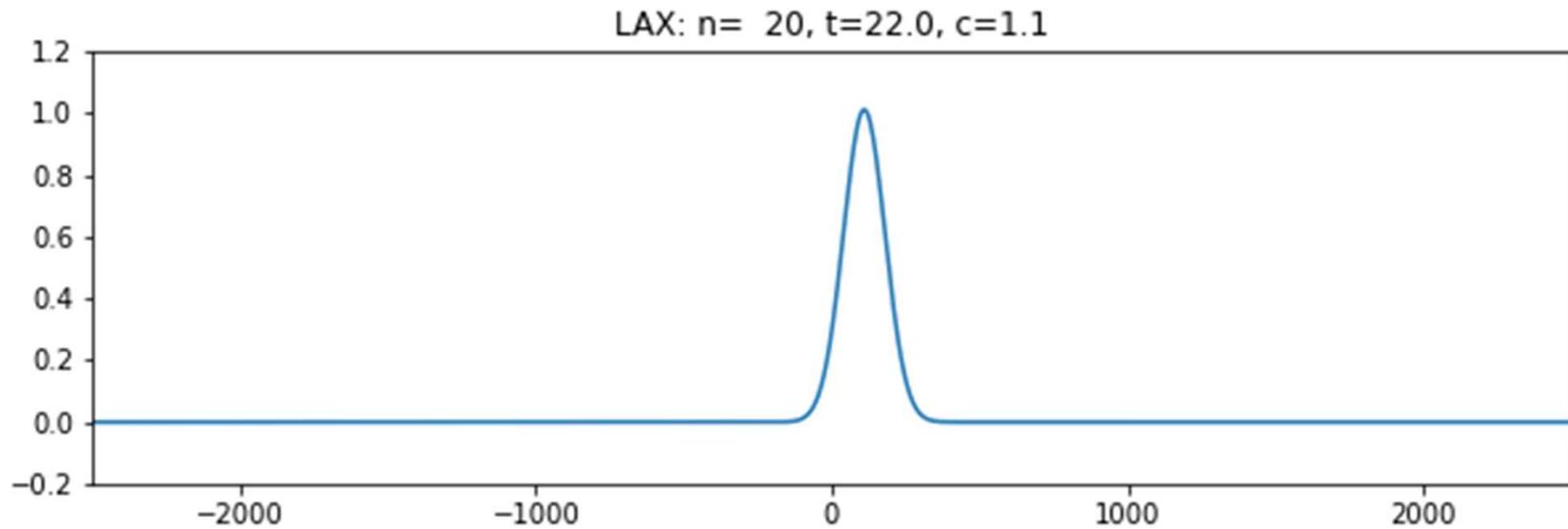
LAX Courant=0.5



LAX Courant=1.0: 5 voltas... perfeito?



LAX Courant=1.1: **instável**



O método de LAX

É **condicionalmente estável**: se $c \leq 1$ ($c = \frac{u\Delta t}{\Delta x}$) c é o **número de Courant** (ou CFL)

É **exato** para $c = 1$. Mas **atenua as pequenas escalas** se $c < 1$. No caso da equação de advecção linear ($u = \text{const}$) é fácil escolher $c = 1$.

Em geral não é possível fazer isso, mas deve evitar-se um c pequeno pois a difusão (numérica) será elevada.

Notem que o $c \leq 1$ pode ser lida como a condição de que, em cada passo de tempo (Δt), a propagação da informação ($l = u\Delta t$) não pode exceder um passo de grelha espacial ($l \leq \Delta x$). **No caso da equação de advecção linear a informação propaga-se à velocidade do fluido (u)**. Noutros casos pode propagar-se mais rapidamente, nomeadamente na forma de uma onda.